

SAN ANTONIO  
**SIGGRAPH**  
≡2002≡

**Integrated Learning for  
Interactive Characters**

**Bruce Blumberg, Marc  
Downie, Yuri Ivanov, Matt  
Berlin, Michael P. Johnson,  
Bill Tomlinson.**

# Practical & compelling real-time learning

- Easy for interactive characters to learn what they ought to be able to learn
- Easy for a human trainer to guide learning process
- A compelling user experience
- Provide heuristics and practical design principles



# Related Work

- **Reinforcement learning**

- Barto & Sutton 98, Mitchell 97, Kaelbling 90, Drescher 91

- **Animal training**

- Lindsay 00, Lorenz & Leyahusen 73, Ramirez 99, Pryor 99, Coppinger 01

- **Motor learning**

- van de Panne et al 93,94, Grzeszczuk & Terzopoulos 95, Hodgins & Pollard 97, Gleicher 98, Faloutsos et al 01

- **Behavior Architectures**

- Reynolds 87, Tu & Terzopoulos 94, Perlin & Goldberg 96, Funge et al 99, Burke et al 01

- **Computer games & digital pets**

- Dogz, AI BO, Black & White

# Dobie T. Coyote Goes to School

QuickTime™ and aAnimation decompressorare needed to see this picture.

# Reinforcement Learning (R.L.) As Starting Point

|    | A1     | A2     | A3     |
|----|--------|--------|--------|
| S1 | Q(1,1) | Q(1,2) | Q(1,3) |
| S2 | Q(2,1) | Q(2,2) | Q(2,3) |
| S3 | Q(3,1) | Q(3,2) | Q(3,3) |

← Set of all possible actions

← Utility of taking action A3 in state S2

↑ Set of all possible states of world

- **Dogs solve a simpler problem in a much larger space & one that is more relevant to interactive characters**

# D.L.: Take Advantage of Predictable Regularities

- **Constrain search for causal agents by taking advantage of temporal proximity & natural hierarchy of state spaces**
  - Use consequences to bias choice of action
  - But vary performance and attend to differences
- **Explore state and action spaces on “as-needed” basis**
  - Build models on demand

# D.L.: Make Use of All Feedback: Explicit & Implicit

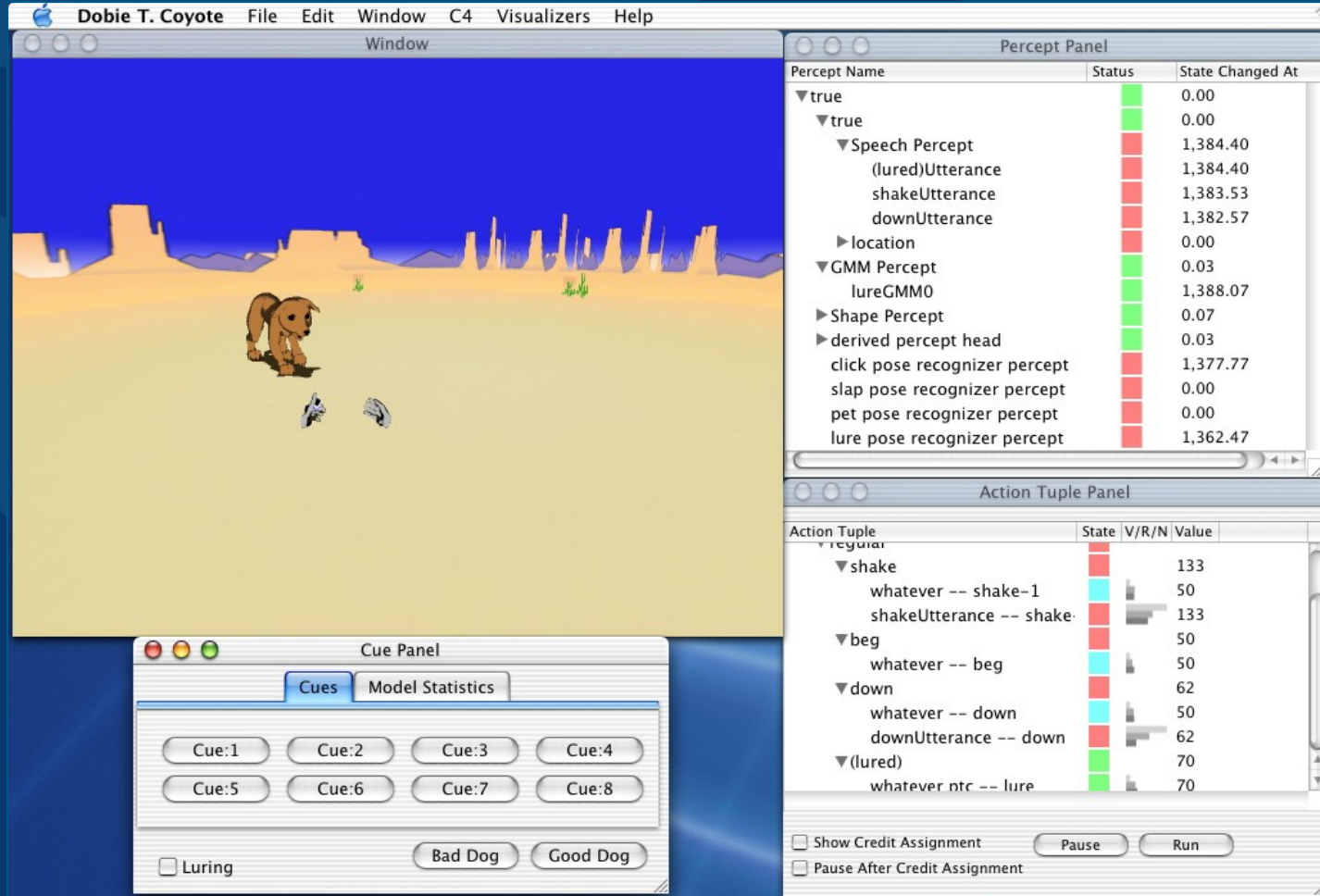
- **Use rewarded action as context for identifying**
  - Promising state space and action space to explore
  - Good examples from which to construct perceptual models, e.g.,
    - A good example of a “sit-utterance” is one that occurs within the context of a rewarded Sit.

# D.L.: Make Them Easy to Train

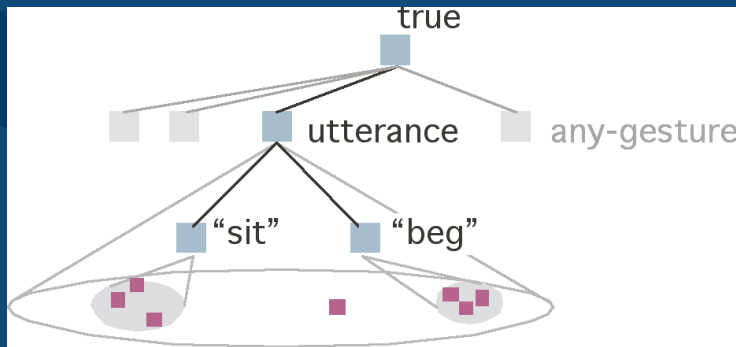
- **Respond quickly to “obvious” contingencies**
- **Support Luring and Shaping**
  - Techniques to prompt infrequently expressed or novel motor actions
- **“Trainer friendly” credit assignment**
  - Assign credit to candidate that matches trainer’s expectation



# The System



# Representation of State: Percept



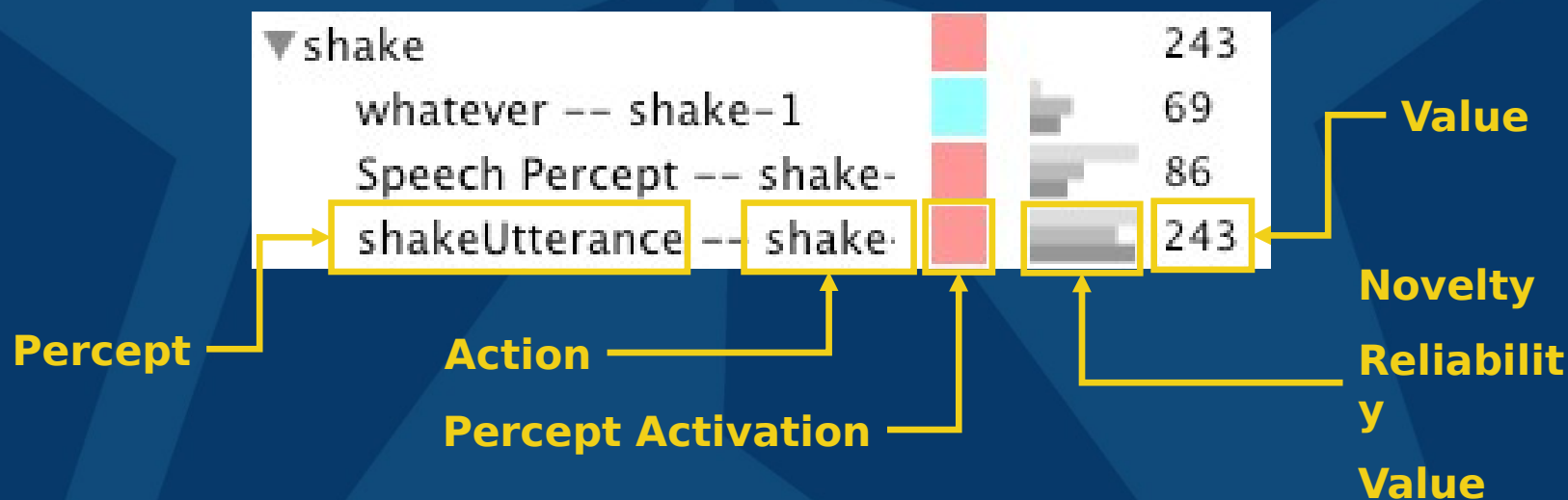
| Percept Name           | Status | State Changed At |
|------------------------|--------|------------------|
| ▼ true                 | Green  | 0.00             |
| ▼ true                 | Green  | 0.00             |
| ▼ Speech Percept       | Red    | 403.67           |
| downUtterance          | Red    | 293.03           |
| shakeUtterance         | Red    | 288.40           |
| begUtterance           | Red    | 296.07           |
| ▶ location             | Red    | 0.00             |
| ▼ GMM Percept          | Green  | 0.03             |
| lureGMM0               | Red    | 434.17           |
| ▶ Shape Percept        | Green  | 0.07             |
| ▶ derived percept head | Green  | 0.03             |

- **Percepts are atomic perception units**
- **Recognize and extract features from sensory data**
- **Model-based**
- **Organized in dynamic hierarchy**

# Representation of State-Action Pairs: Action Tuples

| Action Tuple                | State | V/R/N | Value |
|-----------------------------|-------|-------|-------|
| ▼ Terence's action system   |       |       |       |
| ▼ startle                   |       |       |       |
| heard click -- click!       |       |       | 300   |
| felt slap -- bad dog!       |       |       | 10    |
| rising edge detector:pet -- |       |       | 50    |
| ▼ lure                      |       |       | 140   |
| lure ptc -- lure            |       |       | 140   |
| ▼ regular                   |       |       |       |
| ▼ shake                     |       |       | 243   |
| whatever -- shake-1         |       |       | 69    |
| Speech Percept -- shake-    |       |       | 86    |
| shakeUtterance -- shake     |       |       | 243   |
| ▼ beg                       |       |       | 235   |
| whatever -- beg             |       |       | 63    |
| Speech Percept -- beg       |       |       | 126   |
| begUtterance -- beg         |       |       | 235   |
| ▼ down                      |       |       | 77    |
| whatever -- down            |       |       | 77    |
| ▼ (lured)                   |       |       | 36    |
| whatever ptc -- lure        |       |       | 36    |
| (lured)Utterance -- lure    |       |       | 246   |
| ► navigation                |       |       |       |

Action Tuples are organized in dynamic hierarchy and compete probabilistically based on their learned value and reliability



# Representation of Action: Labeled Path Through Space of Body Configurations

- A motor program generates a path through a graph of annotated poses, e.g.,
  - Sit animation
  - Follow-your-nose procedure
- Paths can be compared and classified just like perceptual events using Motor Model Percepts

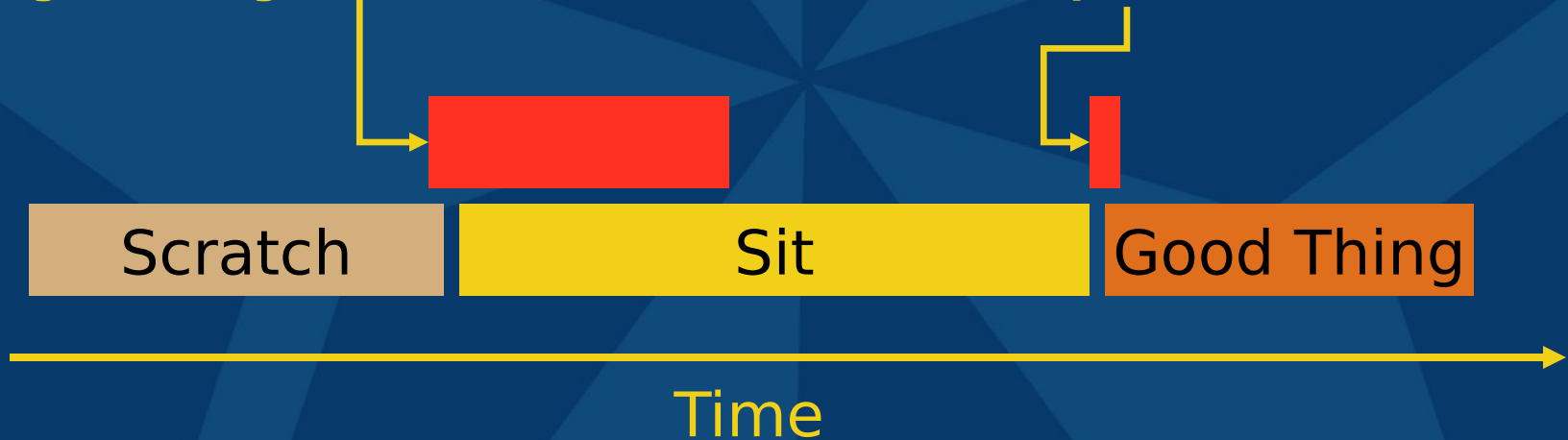
# Use Time to Constrain Search for Causal Agents

## Attention Window:

Look here for cues that appear correlated with increased likelihood of action being followed by a good thing

## Consequences Window:

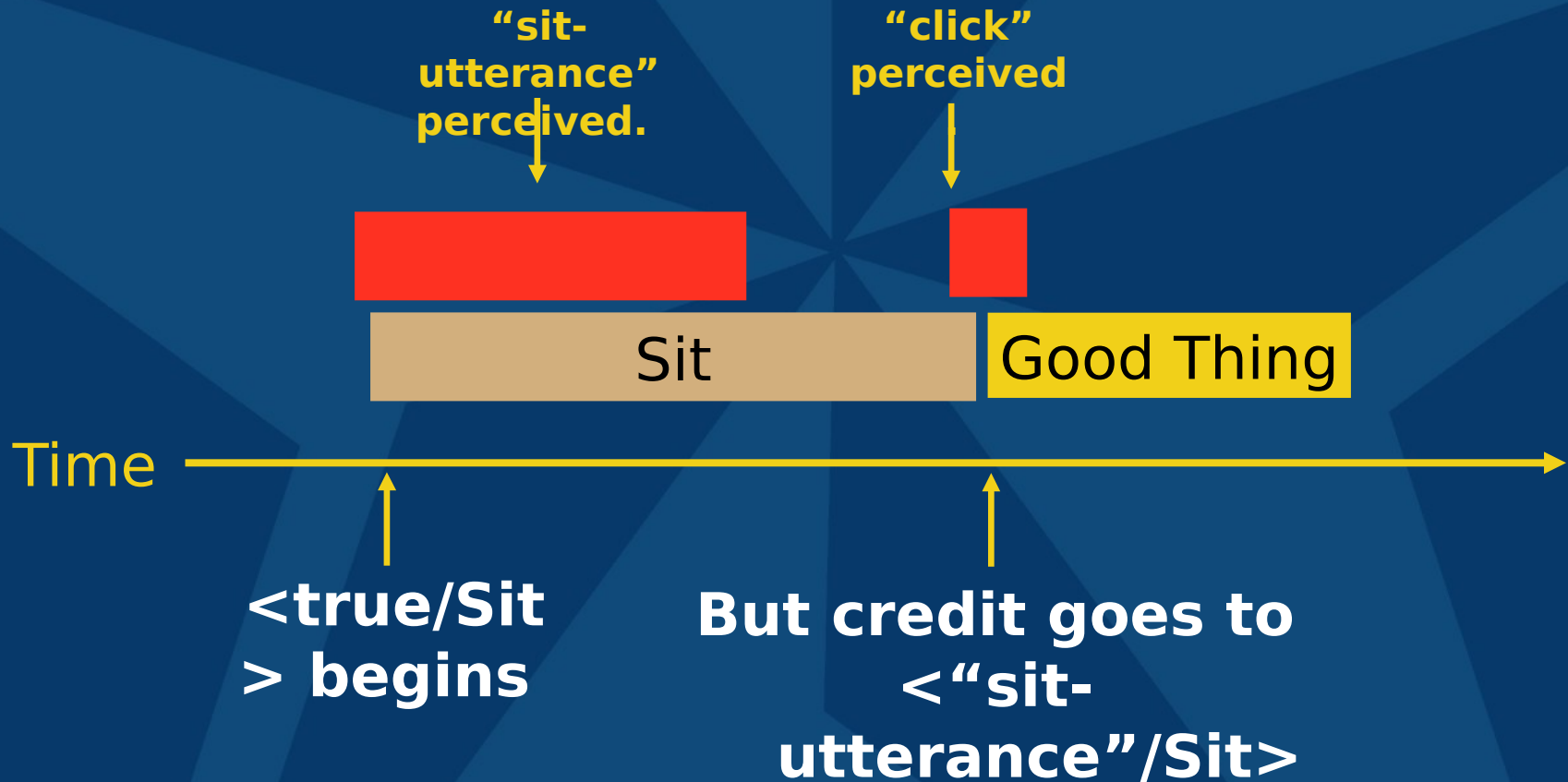
Assume any good or bad things that happen here are associated with the preceding action and the context in which it was performed



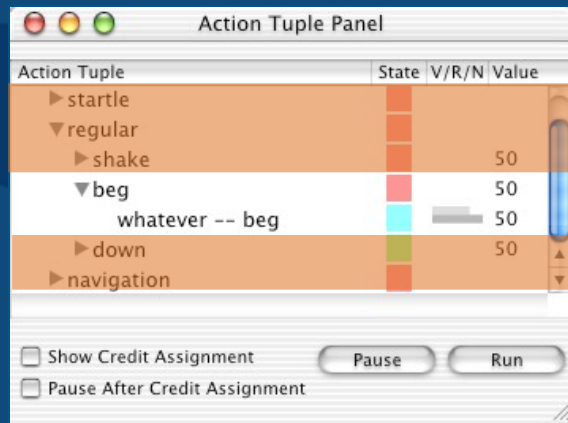
# **Four Important Tasks Are Performed During Credit Assignment**

- **Choose most worthy Action Tuple heuristically based on reliability and novelty statistics**
- **Update value**
- **Create new Action Tuples as appropriate**
- **Guide State and Action Space Discovery**

# Most Worthy Action Tuple Gets Credit



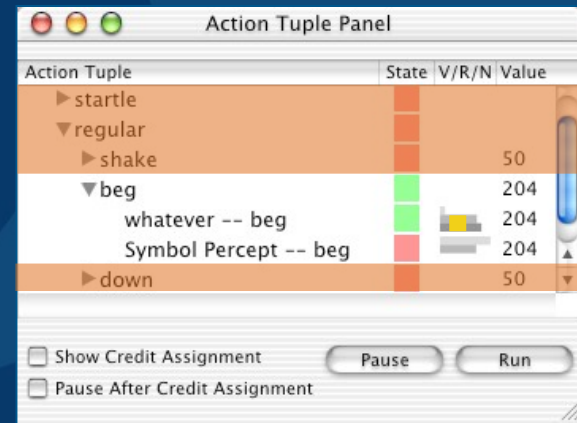
# Create New Action Tuples As Appropriate



Initial state of the Action Tuple Panel:

| Action Tuple    | State | V/R/N | Value |
|-----------------|-------|-------|-------|
| ▶ startle       |       |       |       |
| ▼ regular       |       |       |       |
| ▶ shake         |       |       | 50    |
| ▼ beg           |       |       | 50    |
| whatever -- beg |       |       | 50    |
| ▶ down          |       |       | 50    |
| ▶ navigation    |       |       |       |

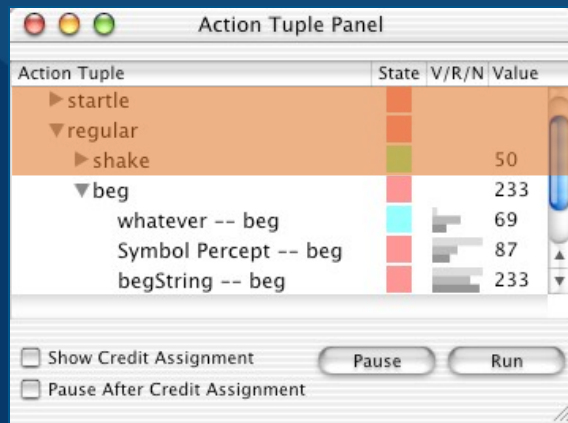
Controls: ☐ Show Credit Assignment, ☐ Pause After Credit Assignment, [Pause], [Run]



First update state of the Action Tuple Panel:

| Action Tuple          | State | V/R/N | Value |
|-----------------------|-------|-------|-------|
| ▶ startle             |       |       |       |
| ▼ regular             |       |       |       |
| ▶ shake               |       |       | 50    |
| ▼ beg                 |       |       | 204   |
| whatever -- beg       |       |       | 204   |
| Symbol Percept -- beg |       |       | 204   |
| ▶ down                |       |       | 50    |

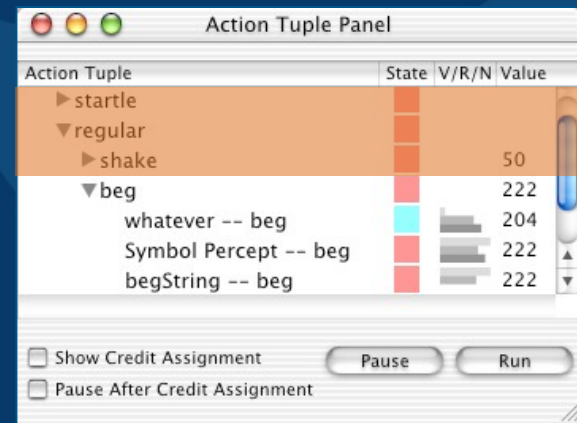
Controls: ☐ Show Credit Assignment, ☐ Pause After Credit Assignment, [Pause], [Run]



Second update state of the Action Tuple Panel:

| Action Tuple          | State | V/R/N | Value |
|-----------------------|-------|-------|-------|
| ▶ startle             |       |       |       |
| ▼ regular             |       |       |       |
| ▶ shake               |       |       | 50    |
| ▼ beg                 |       |       | 233   |
| whatever -- beg       |       |       | 69    |
| Symbol Percept -- beg |       |       | 87    |
| begString -- beg      |       |       | 233   |

Controls: ☐ Show Credit Assignment, ☐ Pause After Credit Assignment, [Pause], [Run]



Third update state of the Action Tuple Panel:

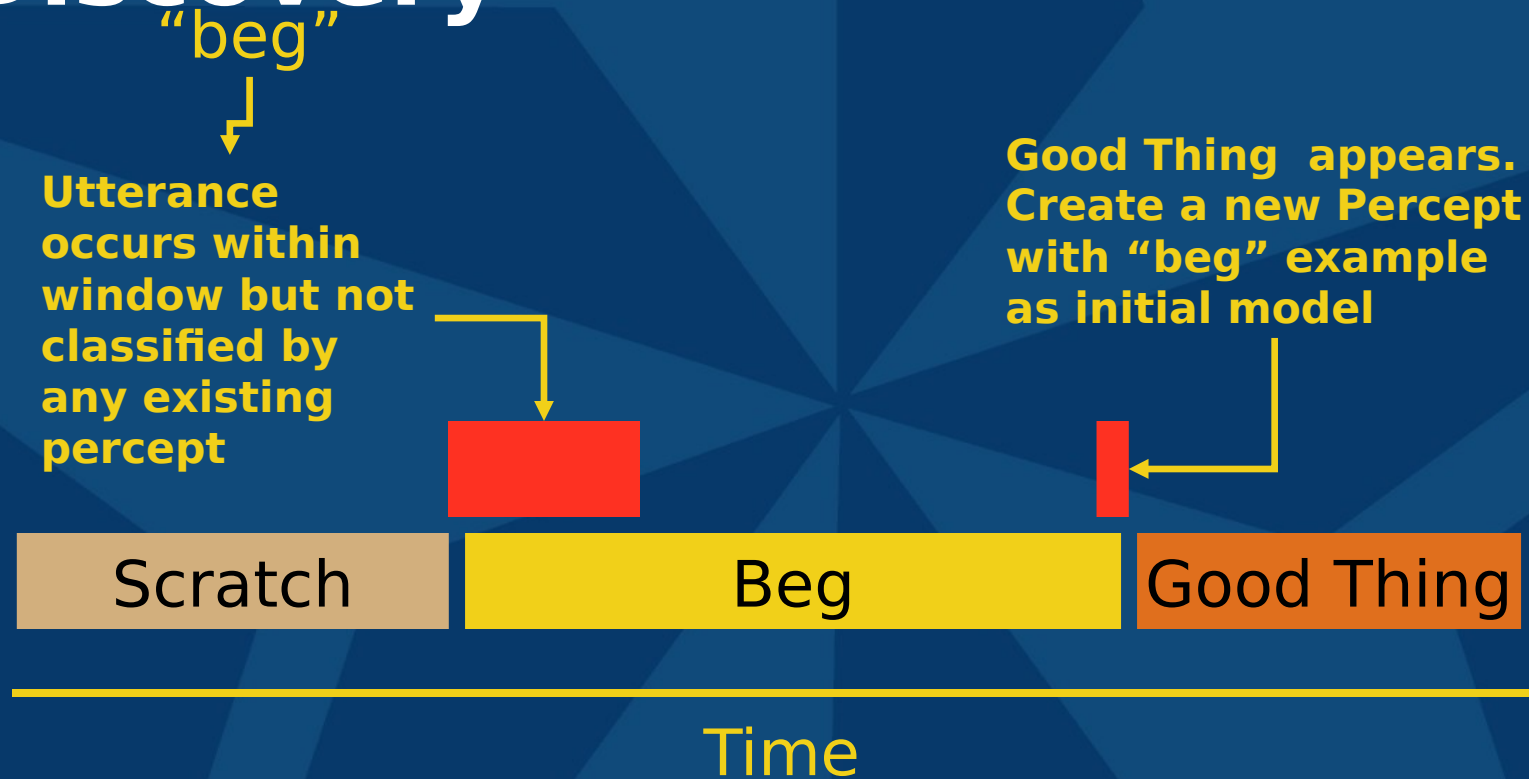
| Action Tuple          | State | V/R/N | Value |
|-----------------------|-------|-------|-------|
| ▶ startle             |       |       |       |
| ▼ regular             |       |       |       |
| ▶ shake               |       |       | 50    |
| ▼ beg                 |       |       | 222   |
| whatever -- beg       |       |       | 204   |
| Symbol Percept -- beg |       |       | 222   |
| begString -- beg      |       |       | 222   |

Controls: ☐ Show Credit Assignment, ☐ Pause After Credit Assignment, [Pause], [Run]



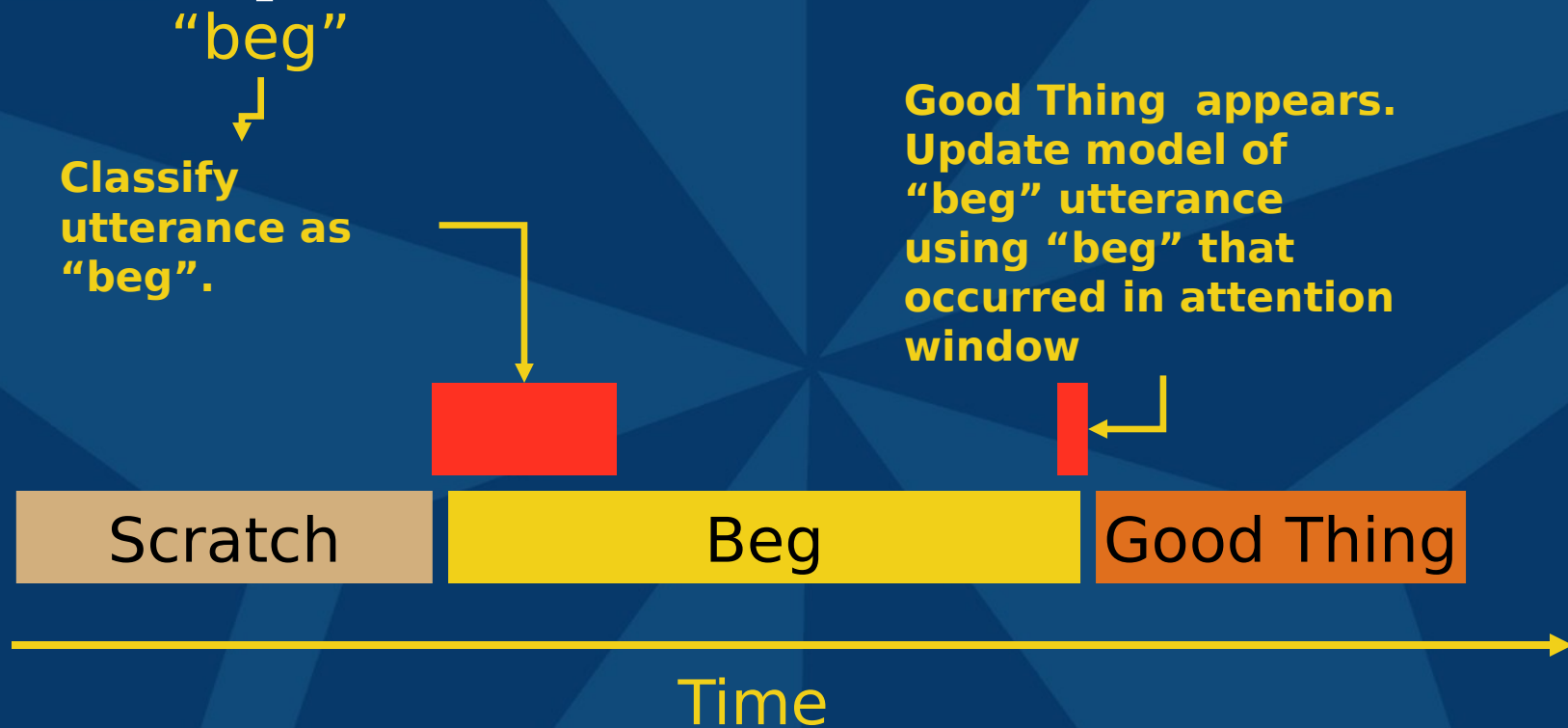


# Implicit Feedback Guides State Space Discovery



This means that Percepts are only created to recognize "promising" utterances

# Implicit Feedback Identifies Good Examples



This means model is built using good examples

# Unrewarded Examples Don't Get Added to Models

"Leg  
"



Utterance classified  
as "Beg" by mistake.  
Beg becomes active.



Scratch

Beg

Sit

Beg ends without food  
appearing. Do not  
update model since  
example may have been  
bad.



Time

Actually, bad examples can be used to  
build model of "not-Beg."

# Implicit Feedback Guides Action Space Discovery

“Follow-your-nose”  
action accumulates  
path through pose-  
space

Good Thing appears.  
Compare accumulated  
path to known paths



Down gets the credit for Good Thing  
appearing, rather than “Follow-your-nose.”

# If Path Is Novel, Create a New Motor Program and Action

“Follow-your-nose”  
action accumulates  
path through pose-  
space

Good Thing appears.  
Compare accumulated  
path to known paths

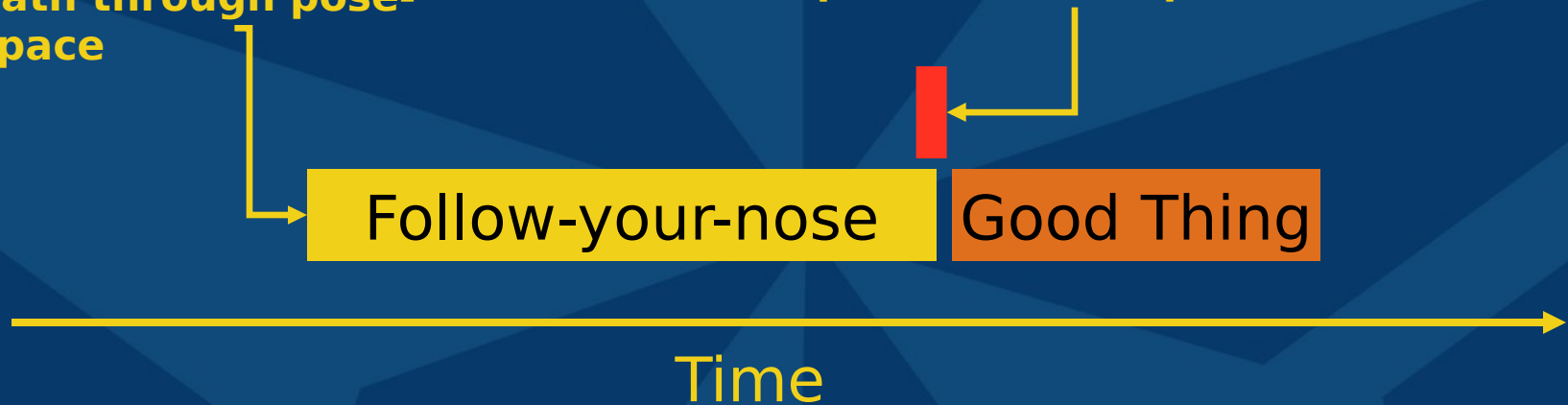


Figure-8

Figure-8 is created and subsequent examples of Figure-8 are used to improve model of path

# Dobie T. Coyote...

QuickTime™ and aAnimation decompressorare needed to see this picture.

# Limitations and Future Work

- **Important extensions**
  - Other kinds of learning (e.g., social or spatial)
  - Generalization
  - Sequences
  - Expectation-based emotion system
- **How will the system scale?**

# Useful Insights

- **Use**
  - Temporal proximity to limit search.
  - Hierarchical representations of state, action and state-action space & use implicit feedback to guide exploration
  - “trainer friendly” credit assignment
- **Luring and shaping are essential**



# Acknowledgements

- **Members of the Synthetic Characters Group, past, present & future**
- **Gary Wilkes**
- **Funded by the Digital Life**  
**m**



SAN ANTONIO

SIGGRAPH

2002

# Reinforcement Learning (R.L.) as starting point

- **Goal**

- Learn optimal set of actions that will take creature from any arbitrary state to a goal state

- **Approach**

- Probabilistically explore states, actions and their outcomes to learn how to act in any given state.